



# KOREAN PATENT ABSTRACTS(KR)

Document Code:A

(11) Publication No.1020000032857 (43) Publication.Date. 20000615

(21) Application No.1019980049474 (22) Application Date. 19981118

(51) IPC Code:

H04N 7/24

(71) Applicant:

KOREA INSTITUTE OF SCIENCE AND TECHNOLOGY

SAMSUNG ELECTRONICS CO., LTD.

(72) Inventor:

CHOI, GEON YEONG

JUNG, UI CHEOL

LEE, JAE HEON

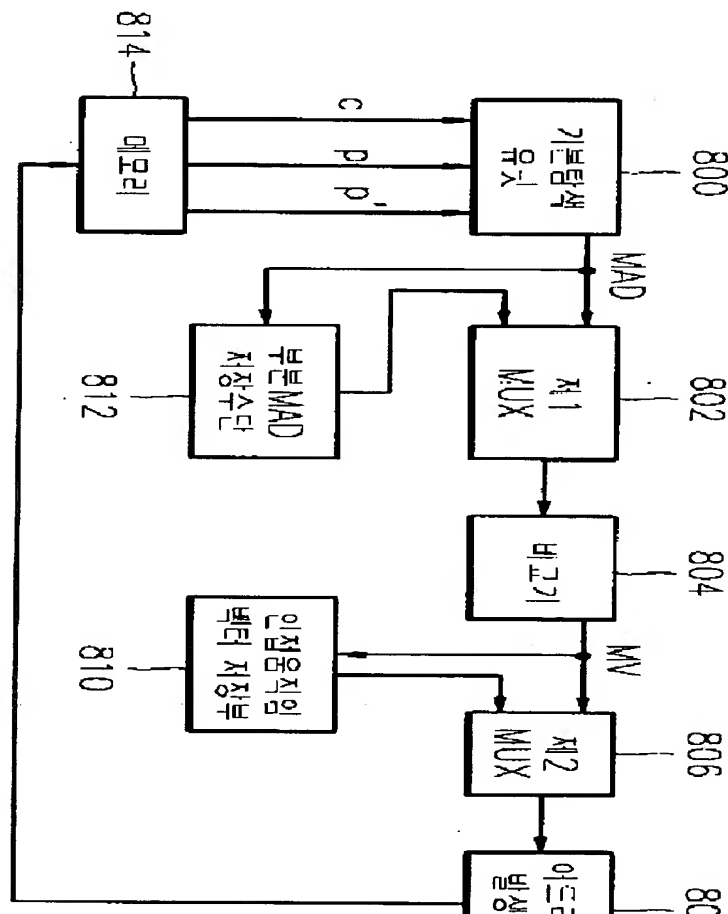
NA, JONG BEOM

(30) Priority:

(54) Title of Invention

DEVICE FOR MOTION ESTIMATION

Representative drawing



(57) Abstract:

PURPOSE: A device for motion estimation is provided so that the mean absolute difference among macro blocks is calculated by repeatedly using one basic searching unit consisting of systolic arrays having a minimum size for processing search blocks and search areas of a minimum size, for thereby making the processing speed higher and making the size of a chip smaller.

CONSTITUTION: A device for motion estimation searches hierarchically upper levels through lower levels when the motion of each macro block of a current frame is estimated by comparison with that of a predetermined search area of a

previous frame, in the case that original images are divided into images of a lower level(104), images of an intermediate level(102), and images of an upper level(100). A memory(814) stores current frame data and previous frame data. A basic searching unit(800) reads block data of a predetermined size of the current frame and search area data of the previous frame from the memory and outputs the matching criterion value of these data. A partial MAD storing unit(812) divides a search area into a block size, and stores and outputs an output value of the basic searching unit upon receipt of data in each divided search area and the block data, which is the value obtained by summing a previously stored value, in case of search at the lower level. A comparison unit(804) outputs an index of a corresponding searching point in the search area which is outputted from the partial MAD storing unit and has a minimum value by sequential comparison, wherein a searching is achieved at the lower level (104). An address generation unit(808) obtains a motion vector from the index when the searching is achieved at the lower level(104).

COPYRIGHT 2000 KIPO

if display of image is failed, press (F5)

(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) Int. Cl. <sup>6</sup> H04N 7/24		(45) 공고일자 (11) 등록번호 (24) 등록일자	2003년01월24일 10-0359091 2002년10월17일
(21) 출원번호 (22) 출원일자 (73) 특허권자	10-1998-0049474 1998년11월18일 삼성전자 주식회사	(65) 공개번호 (43) 공개일자	특2000-0032857 2000년06월15일
(72) 발명자	나종범 대전광역시 유성구 전민동 464-1 엑스포아파트 404동 506호 이재현 대전광역시 유성구 구성동 373-1 정의철 경기도 안양시 만안구 박달2동 신한아파트 102동 902호 최건영 경기도 수원시 팔달구 매탄동 1162 임광아파트 10동 110호		
(74) 대리인	이영필, 권석홍, 이상용		

심사관 : 조광현

(54) 움직임추정장치

요약

본 발명은 움직임 추정 장치에 관한 것으로, 현재 프레임의 각 매크로 블록의 움직임을 상기 매크로 블록에 대응하는 이전 프레임의 탐색영역과 비교하여 추정할 때, 각 프레임에 대해 계층적으로 탐색하는 움직임 추정 장치에 있어서, 현재 프레임 데이터와 이전 프레임 데이터를 저장하는 메모리; 메모리로부터 매크로 블록 데이터를 소정 크기의 블록단위로 독출하고, 탐색 영역 데이터를 독출하여 독출된 데이터들의 정합 정도를 나타내는 정합기준값을 출력하는 기본 탐색 유닛; 중간레벨 또는 하위레벨인 경우 기본 탐색유닛의 출력값을 매크로 블록의 전 영역에 대해 가산하여 저장하는 부분 정합기준값 저장수단; 기본 탐색 유닛의 출력값과 부분 정합 기준값 저장수단의 출력값중 하나를 선택하여 출력하는 제1멀티플렉서; 제1멀티플렉서의 출력값을 순차적으로 비교하여 탐색영역내에서 가장 작은 정합기준값을 갖는 탐색점의 인덱스를 출력하는 비교기; 하위레벨을 탐색하는 경우 현재 처리중인 매크로 블록의 인접 매크로 블록중 움직임 벡터값을 이미 알고있는 매크로 블록에 대한 움직임 벡터들의 중간값을 저장하는 인접움직임 벡터 저장부; 비교기의 출력과 인접움직임 벡터 저장부의 출력중 하나를 출력하는 제2멀티플렉서; 및 제2멀티플렉서에서 출력되는 인덱스에 해당하는 메모리의 어드레스를 발생하는 어드레스 발생부를 포함함을 특징으로한다.

대표도

도8

명세서

도면의 간단한 설명

- 도 1은 본 발명에 사용되는 계층적 탐색 알고리즘의 계층도를 도시한 것이다.
- 도 2는 현재 탐색중인 매크로 블록의 인접 매크로 블록들을 도시한 것이다.
- 도 3은 기본 탐색 유닛의 상세도이다.
- 도 4는 기본 탐색 유닛을 상위레벨에 적용하는 경우를 도시한 것이다.
- 도 5는 기본 탐색 유닛을 중간레벨에 적용하는 경우를 도시한 것이다.
- 도 6은 기본 탐색 유닛을 하위레벨에 적용하는 경우를 도시한 것이다.
- 도 7은 중간레벨과 하위레벨의 탐색시 발생하는 부분 MAD를 저장하는 수단을 도시한 것이다.
- 도 8은 본 발명에 따른 움직임 추정 장치에 대한 블록도이다.

발명의 상세한 설명

## 발명의 목적

### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 움직임 추정 장치에 관한 것이다.

더욱 더 치열해지는 현 사회에서 좀 더 정확한 정보의 전달과 공유를 위하여 점점 중요시되는 수단이 영상통신이다. 특히 실시간 동영상은 좀 더 정확하고 사실적인 정보의 전달매체로 사용된다. 실시간 디지털 텔레비전, 동영상 스토리지 디바이스, 동영상 원격회의, 영상 전화기 등은 영상을 정보전달 매개체로서 현장감을 살릴 수 있게 한 것들이다. 이들 동영상 칩을 구현하는데 있어서 가장 많은 계산량을 필요로 하는 것이 동영상 부호화를 위한 움직임 추정(Motion Estimation)부분이다. 움직임 추정은 종종 칩의 전 처리 파워의 50%까지도 차지한다.

지금까지 제안된 여러가지 움직임 추정 알고리즘중 전역 탐색기법(Full Search Block Matching Algorithm)의 경우 실시간 구현시 매우 큰 계산량으로 인해 칩 크기가 현저히 커지는 단점이 있다. 이런 단점을 보완하기 위하여 여러 가지 탐색 알고리즘이 제안되어 왔다. 하지만 이런 고속 탐색 알고리즘들은 계산량에는 큰 이득을 주었으나 전역 탐색기법에 비해 성능이 많이 떨어진다.

고속 탐색 알고리즘들중 계층적 탐색 알고리즘은 계산량이 현저히 줄어들어도 불구하고 성능은 전역 탐색에 비해 크게 떨어지지 않는다. 계층적 탐색 알고리즘의 경우, 각 단계마다 처리하는 매크로 블록의 크기와 탐색 영역의 크기가 서로 다르다. 일반적으로, 움직임 추정 알고리즘을 구현할 때, 시스템릭 어레이(Systolic Array)를 사용하게되는데, 시스템릭 어레이의 크기는 처리하는 블록과 탐색영역의 크기에 따라 결정된다.

예를 들어, 3단 계층적 탐색 알고리즘을 생각해 보자. 우선 매크로 블록의 크기가 16x16인 경우, 상위 계층에서는 가로, 세로가 각각 1/4씩 줄어든 영상을 이용하여 4x4블록 단위로  $\pm p_0$ 영역에 대해 탐색을 수행한다. 이전 프레임의 탐색 영역의 후보 블록들 중 현재 프레임에서 현재 처리중인 블록과 가장 유사한 블록을 찾는다. 중간 계층에서는 상위계층에서 찾아진 위치를 초기점으로하여 다시 매크로 블록을 가로, 세로 각각 1/2씩 줄인 영상을 이용하여 8x8블록 단위로  $\pm p_1$ 영역에 대해서 탐색을 다시 수행한다. 마찬가지로 중간계층에서는 이렇게 상위계층에서 찾아진 위치를 초기점으로하여 마지막으로 하위레벨에서 16x16 크기의 매크로블록 단위로  $\pm p_2$ 영역에 대해 탐색을 수행하여 최종 움직임 벡터를 구하게된다. 따라서, 계층적 탐색 알고리즘은 상술한 바와 같이 각 단계마다 서로 다른 크기의 블록이 사용되며, 이들 블록을 위해 3가지 서로 다른 크기의 시스템릭 어레이들이 필요하다.

그러나, 계층적 탐색은 순차적으로 수행되므로, 어느 한 시점에서 살펴보면 하나의 시스템릭 어레이만 동작하고 나머지는 아무런 작업도 수행하지 않는다. 따라서 이 구조는 필요이상의 시스템릭 어레이를 사용하여 칩 면적을 증가시키는 문제점이 있다.

### 발명이 이루고자하는 기술적 과제

본 발명이 이루고자하는 기술적 과제는 움직임 추정시 최소 크기의 탐색 블록과 탐색 영역을 처리하는 최소의 크기를 가지는 시스템릭 어레이로 이루어진 하나의 기본 탐색 단위를 반복적으로 이용하여 움직임 벡터를 찾는 움직임 추정 장치를 제공하는데 있다.

### 발명의 구성 및 작용

상기 기술적 과제를 이루기 위한, 본 발명은 원 영상을 하위레벨 영상, 원 영상에 대해 각각 가로, 세로로 1/2 크기인 영상을 중간레벨 영상 및 원 영상에 대해 각각 가로, 세로로 1/4 크기인 영상을 상위레벨 영상으로하여 현재 프레임의 각 매크로 블록의 움직임을 상기 매크로 블록에 대응하는 이전 프레임의 탐색영역과 비교하여 추정할 때, 상기 상위 레벨로부터 상기 하위 레벨까지 계층적으로 탐색하는 움직임 추정 장치에 있어서, 현재 프레임 데이터와 이전 프레임 데이터를 저장하는 메모리; 상기 메모리로부터 상기 매크로 블록 데이터를 소정 크기의 블록단위로 독출하고, 상기 탐색 영역 데이터를 독출하여 독출된 데이터들의 정합 정도를 나타내는 정합기준값을 출력하는 기본 탐색 유닛; 상기 중간레벨 또는 하위 레벨을 탐색하는 경우 상기 기본 탐색유닛의 출력값을 상기 매크로 블록의 전 영역에 대해 가산하여 저장하는 부분 정합기준값 저장수단; 탐색 레벨에 따라 상기 기본 탐색 유닛의 출력값과 상기 부분 정합기준값 저장수단의 출력값중 하나를 선택하여 출력하는 제1멀티플렉서; 상기 제1멀티플렉서로부터 출력되는 값을 순차적으로 비교하여 상기 탐색영역내에서 가장 작은 정합기준값을 갖는 탐색점의 인덱스를 출력하는 비교기; 상기 하위레벨을 탐색하는 경우 현재 처리중인 매크로 블록의 인접 매크로 블록중 움직임 벡터값을 이미 알고있는 매크로 블록에 대한 움직임 벡터들의 중간값을 저장하는 인접움직임 벡터 저장부; 탐색 레벨에 따라 상기 비교기의 출력과 상기 인접움직임 벡터 저장부의 출력중 하나를 출력하는 제2멀티플렉서; 및 상기 제2멀티플렉서에서 출력되는 인덱스에 해당하는 상기 메모리의 어드레스를 발생하여 상기 메모리에 출력하는 어드레스 발생부를 포함함을 특징으로한다.

이하에서 첨부된 도면을 참조하여 본 발명을 실시예를 보다 상세히 설명하기로 한다. 도 1은 본 발명에 사용되는 계층적 탐색 알고리즘(Hierarchical Search Block Matching Algorithm by using 3 candidates and Spatial correlation, 이하 HSBMA3S라 함)의 계층도를 도시한 것이다. 도 1에 따른 알고리즘은 상위(100), 중간(102) 및 하위 레벨(104)로 구성된다.

하위레벨(104)의 영상은 원 영상이고, 중간레벨(102)의 영상은 원 영상에 대해 가로, 세로로 각각 1/2 크기의 영상이다. 상위레벨(100)의 영상은 중간레벨(102)의 영상을 서브샘플링한 것으로 원 영상에 대해 가로, 세로가 각각 1/4인 영상이다. 이러한 영상에 대해 먼저, 상위레벨(100)에서 전역 탐색을 수행하고, 중간레벨(102)과 하위레벨(104)에서는 이전 레벨에서 얻은 움직임 벡터를 초기점으로 각각 순차적

으로 부분 탐색을 수행하여 좀 더 세밀하게 조정한 후 최종 움직임 벡터를 얻게된다.

상위레벨(100)에서는 가로, 세로 각각 원 영상의 1/4크기인 영상을 이용하여 탐색을 수행한다. 가로, 세로 각각 1/4씩 줄어든 4x4크기의 블록단위로 탐색영역의 크기도 가로, 세로 각각 1/4씩 줄인 수평/수직  $\pm 4$  화소 영역에 대해서 전역 탐색을 수행한다. 정합 기준(Matching Criterion)은 평균 절대값 차(Mean Absolute Difference, 이하 MAD라 함)이다. 상위레벨(100)의 탐색 후, MAD가 최소인 두 점을 중간레벨(102)의 탐색을 위한 움직임 벡터 후보로 선택한다.

중간레벨(102)에서는 가로, 세로 각각 원 영상의 1/2 크기인 영상을 이용하여 탐색한다. 중간레벨(102)에서는 총 3개의 움직임 벡터 후보에 대해서, 각각의 움직임 벡터 후보를 초기 탐색점으로하여 8x8크기의 블록 단위로 수평/수직  $\pm 2$  화소 크기의 부분 탐색을 수행한다. 3개의 초기 탐색점중 2개는 상위레벨(100)에서 MAD를 정합기준으로하여 선택된 것이고, 나머지 하나는 하위레벨(104)에서 움직임 벡터들의 공간적 상관관계(Spatial Correlation)를 이용하여 선택된다. 중간레벨(102)의 3개의 후보 탐색점에 대해서 탐색을 수행하여 최소 MAD를 가지는 점을 각각 하나씩 얻게되면, 그 중에 MAD가 최소인 한 점을 하위레벨(104)에서의 초기 탐색점으로 선택한다.

도 2는 하위레벨(104)로부터 중간레벨(102)의 3개의 후보 탐색점중 하나를 얻는 것을 설명하기위한 도면으로, 하위레벨(104)에서 현재 처리중인 매크로 블록의 인접 매크로 블록들을 도시한 것이다. 인접 매크로 블록들은 각각 NMV0, NMV1, NMV2이다. 이들 매크로 블록들은 시간적으로 우선 처리되어있어서 이미 최종 움직임 벡터들(NMV0<sub>x</sub>, NMV0<sub>y</sub>), (NMV1<sub>x</sub>, NMV1<sub>y</sub>), (NMV2<sub>x</sub>, NMV2<sub>y</sub>)을 알고있는 블록들이다. 이들의 움직임 벡터들의 x, y성분을 각각 따로 중간값(median)을 취해 하나의 후보로 결정한다.

$$MV_x = \text{Median}\{NMV0_x, NMV1_x, NMV2_x\}$$

$$MV_y = \text{Median}\{NMV0_y, NMV1_y, NMV2_y\}$$

이렇게 얻어진 움직임 벡터 후보를 1/2로 크기를 줄여서 중간 레벨에서의 초기 탐색점으로 사용한다.

하위레벨(104)에서는 원 영상을 그대로 이용하여 중간 레벨(102)에서 얻어진 점을 중심으로 16x16 크기의 블록단위로 수평/수직  $\pm 2$  화소의 부분탐색을 수행하여 최종 움직임 벡터를 하나 결정하게 된다.

상술한 HSBMA3S를 하드웨어로 구현하는 본 발명은 시스틀릭 어레이를 기반으로한다. 시스틀릭 어레이는 MAD를 계산하기 위한 간단한 연산(별셈, 절대값, 덧셈)을 수행하는 처리 단위(Processing Element, 이하 PE라 함)들의 배열이다. 일반적으로 PE의 개수는 움직임 추정 알고리즘에서 처리하는 매크로 블록의 크기 또는 탐색 영역의 크기와 비례한다. 예를 들어, 전역 탐색의 경우 16x16 매크로블록 단위로 처리할 경우 256개의 PE가 사용되며 움직임 추정 칩의 대부분을 PE가 차지하게 되어 칩 사이즈가 커지게 된다. HSBMA3S 알고리즘은 3개의 레벨로 이루어져 있고 각각의 레벨마다 처리하는 블록의 크기와 탐색 영역의 크기가 서로 다르다. 따라서 각 레벨에 적합한 서로 다른 시스틀릭 어레이가 필요하게 된다. 본 발명에서는 모든 레벨에서 공통적으로 사용할 수 있는 하나의 시스틀릭 어레이로 기본 탐색 유닛 (Basic Searching Unit, 이하 BSU라 함)을 두어 이를 반복적으로 사용함으로써 움직임 추정을 수행한다. 그리고 BSU 또한 5개의 PE로 이루어진 구조이므로 칩 전체에서 PE가 차지하는 비율은 미미하다.

HSBMA3S는 상위 레벨, 중간 레벨, 하위 레벨에서 처리하는 블록의 크기와 탐색 영역의 크기가 서로 다르다. 상위 레벨에서는 4x4 크기의 블록에 대해서 수평/수직  $\pm 4$  화소 탐색을 수행하고, 중간 레벨에서는 8x8 크기의 블록에 대해서  $\pm 2$  화소 탐색, 하위 레벨에서는 16x16 크기의 블록에 대해서  $\pm 2$  화소 탐색을 수행한다. 각 레벨에서 처리하는 블록의 크기 중 가장 작은 크기와 탐색 영역의 크기 중 가장 작은 것을 택하여 4x4 크기의 블록에 대해서  $\pm 2$  화소 탐색을 수행할 수 있는 BSU를 둔다. BSU를 각 레벨에서 반복적으로 사용함으로써 탐색을 수행하게 된다.

도 3은 BSU의 상세도이다. BSU는 4x4 크기의 블록에 대하여  $\pm 2$  화소의 탐색을 수행하는 시스틀릭 어레이이다. 도 3에 따르면, BSU는 5개의 PE(PE0, PE1, PE2, PE3, PE4), 복수의 플립플롭(OFF) 및 멀티플렉서(MUX)로 구성된다.

BSU가 4x4 크기의 블록에 대하여  $\pm 2$  화소의 탐색을 수행하기 위해서는 다음의 표 1 및 표 2와 같이 현재 프레임의 4x4 크기의 기준 블록 데이터와 이전 프레임의 8x8 크기의 탐색 영역 데이터들이 필요하다.

[표 1]

R(0,0)	R(0,1)	R(0,2)	R(0,3)
R(1,0)	R(1,1)	R(1,2)	R(1,3)
R(2,0)	R(2,1)	R(2,2)	R(2,3)
R(3,0)	R(3,1)	R(3,2)	R(3,3)

[표 2]

S(0,0)	S(0,1)	S(0,2)	S(0,3)	S(0,4)	S(0,5)	S(0,6)	S(0,7)
S(1,0)	S(1,1)	S(1,2)	S(1,3)	S(1,4)	S(1,5)	S(1,6)	S(1,7)
S(2,0)	S(2,1)	S(2,2)	S(2,3)	S(2,4)	S(2,5)	S(2,6)	S(2,7)
S(3,0)	S(3,1)	S(3,2)	S(3,3)	S(3,4)	S(3,5)	S(3,6)	S(3,7)
S(4,0)	S(4,1)	S(4,2)	S(4,3)	S(4,4)	S(4,5)	S(4,6)	S(4,7)
S(5,0)	S(5,1)	S(5,2)	S(5,3)	S(5,4)	S(5,5)	S(5,6)	S(5,7)
S(6,0)	S(6,1)	S(6,2)	S(6,3)	S(6,4)	S(6,5)	S(6,6)	S(6,7)
S(7,0)	S(7,1)	S(7,2)	S(7,3)	S(7,4)	S(7,5)	S(7,6)	S(7,7)

기준 블록 데이터는 C 포트를 통해, 탐색 영역 데이터는 P와 P'의 두 개의 포트로 나뉘어 BSU로 입력된다. 탐색영역 데이터는 상기 두 포트에 동시에 입력되며, 실질적으로 PE에 입력되는 순서는 '리셋'단자에 연결된 DFF들과 MUX들에 의해 결정된다. 탐색 영역 데이터가 두 포트를 통해서 나누어서 동시에 입력하는 이유는 데이터의 공급을 원활히 하여 PE의 효율성을 극대화하기 위함이다. 탐색 영역 데이터 64개 중 왼쪽 네 개 열의 데이터 32개는 P 포트를 통해서 오른쪽 네 개 열의 데이터 32개는 P' 포트를 통해서 동시에 입력된다.

BSU에 입력된 기준 블록 데이터는 각 플립플롭(DFF)을 거쳐서 5개의 PE에 순차적으로 공급되고, 탐색 영역 데이터는 5개의 PE에 브로드캐스팅된다. PE0는 P를, PE4는 P'을, 나머지 PE들(PE1, PE2, PE3)은 각 MUX를 통해 P와 P'중 적절한 데이터를 선택하여 공급받게 된다. PE는 4x4 크기의 블록 데이터에 대한 MAD를 계산하기 위한 연산을 담당한다.

BSU는  $\pm 2$  화소, 즉 수평/수직 5개씩 25개의 탐색점에 대한 탐색을 행별로 순차적으로 수행한다. 5개의 PE는 하나의 행의 5개의 탐색점에 각각 할당되어 그 탐색점에서의 MAD를 계산한다. 한번에 하나의 행씩 5개의 행에 위치하는 탐색점들에 대한 탐색을 수행하면 하나의 기본 탐색 단위인 4x4 크기의 블록에 대한  $\pm 2$  화소의 탐색을 마치게 된다. 각각의 PE에 입력되는 기준 블록 데이터와 탐색 영역 데이터는 표 3과 같다. 표에서는 9번째 클럭 사이클(CC)까지만 나타낸 것이다.

[표 3]

클럭 사이클	PE0		PE1		PE2		PE3		PE4	
	RD	SD	RD	SD	RD	SD	RD	SD	RD	SD
0	R(0,0)	S(0,0)								
1	R(0,1)	S(0,1)	R(0,0)	S(0,1)						
2	R(0,2)	S(0,2)	R(0,1)	S(0,2)	R(0,0)	S(0,2)				
3	R(0,3)	S(0,3)	R(0,2)	S(0,3)	R(0,1)	S(0,3)	R(0,0)	S(0,3)		
4	R(1,0)	S(1,0)	R(0,3)	S(0,4)	R(0,2)	S(0,4)	R(0,1)	S(0,4)	R(0,0)	S(0,4)
5	R(1,1)	S(1,1)	R(1,0)	S(1,1)	R(0,3)	S(0,5)	R(0,2)	S(0,5)	R(0,1)	S(0,5)
6	R(1,2)	S(1,2)	R(1,1)	S(1,2)	R(1,0)	S(1,2)	R(0,3)	S(0,6)	R(0,2)	S(0,6)
7	R(1,3)	S(1,3)	R(1,2)	S(1,3)	R(1,1)	S(1,3)	R(1,0)	S(1,3)	R(0,3)	S(0,7)
8	R(2,0)	S(2,0)	R(1,3)	S(1,4)	R(1,2)	S(1,4)	R(1,1)	S(1,4)	R(1,0)	S(1,4)
9	R(2,1)	S(2,1)	R(2,0)	S(2,1)	R(1,3)	S(1,5)	R(1,2)	S(1,5)	R(1,1)	S(1,5)

여기서, RD는 기준 블록 데이터이고, SD는 탐색 영역 데이터이다.

기준 블록 데이터와 탐색 영역 데이터는 각각 행별로 차례대로 입력되며 P'을 통해 입력되는 탐색 영역 데이터는 C와 P를 통해 입력되는 데이터들보다 네 클럭 늦게 입력된다. 기준 블록 데이터는 C 포트를 통해 16 클럭마다 반복되어 재입력되며, 탐색 영역 데이터들은 16개의 데이터가 입력되고 나면 한 행씩 일의 데이터가 16개 입력되는 방식으로 진행된다.

하나의 PE에서 4x4 블록에 대한 MAD를 계산하는데 16CC이 걸리고, 5개의 PE에서 동시에 5개의 탐색점들에 대한 탐색이 수행하므로,  $\pm 2$  화소, 25개의 탐색점들에 대한 탐색을 수행하는 데는 총 16x5=80CC이 소요된다.

도 4는 BSU를 상위레벨에 적용하는 경우를 도시한 것이다. 상위 레벨은 4x4 크기의 블록에 대하여  $\pm 4$  화소 탐색을 수행한다. 처리하는 블록의 크기는 BSU가 처리하는 블록의 크기와 동일하나 탐색 영역의 크기는 다르다. 따라서, BSU를 이용하여 탐색을 수행하기 위해서 도 4와 같이 탐색점들을 네 영역으로 나누어 처리한다. BSU는 한번에  $\pm 2$  화소, 즉 수평/수직 5개씩 25개 탐색점들에 대한 탐색을 수행하므로 탐색점들을 25개씩 나누어 순차적으로 처리한다.

도 5는 BSU를 중간레벨에 적용하는 경우를 도시한 것이다. 도시된 바에 따르면, 중간 레벨은 8x8 크기의

블록에 대하여  $\pm 2$  화소 부분탐색을 수행한다. 탐색 영역의 크기는 BSU가 처리하는 것과 동일하고 처리하는 블록의 크기는 다르다. BSU를 이용하여 탐색을 수행하기 위해서 도시된 바와 같이 처리하는 블록을 네 개의 4x4 크기의 블록으로 나누어 순차적으로 처리한다. 이때 BSU에서 계산된 MAD는 8x8 블록에 대한 완전한 MAD가 아니라 4x4 블록에 대한 부분 MAD이다. 부분 MAD들을 저장하여 완전한 MAD를 구하는 추가 하드웨어가 필요하다.

도 6은 BSU를 하위레벨에 적용하는 경우를 도시한 것이다. 도시된 바에 따르면, 하위 레벨에서는 16x16 크기의 블록에 대하여  $\pm 2$  화소 부분 탐색을 수행한다. 중간 레벨과 마찬가지로 처리하는 탐색 영역의 크기는 BSU가 처리하는 크기와 동일하고 처리하는 블록의 크기는 서로 다르다. 중간 레벨에서와 같은 방법으로 처리하는 블록을 16 개의 4x4 크기의 블록으로 나누어 순차적으로 처리한다. 여기서도 완전한 MAD를 구하기 위한 하드웨어를 사용한다.

도 7은 중간레벨과 하위레벨의 탐색시 발생하는 부분 MAD를 저장하는 수단을 도시한 것이다. 중간 레벨과 하위 레벨에서는 처리하는 블록을 4x4 크기의 블록으로 나누어 처리하기 때문에 이러한 모듈이 사용된다. 도시된 바에 따르면, 부분 MAD 저장수단은 25개의 쉬프트 레지스터(700)와 덧셈기(702)로 이루어져 있다. BSU는 한번에  $\pm 2$  화소의 탐색, 즉 25개 점에 대한 탐색을 수행하므로 각각의 탐색점들에서 얻어진 부분 MAD를 저장할 공간으로 25개의 쉬프트 레지스터(700)가 사용된다. 탐색영역이  $\pm p_2$ 라면, 쉬프트 레지스터(700)의 개수는  $(2p_2+1)^2$  개가 필요하다. 16x16 블록에 대한 완전한 MAD는 16 비트이므로 각각의 레지스터의 크기는 16비트가 된다. 중간 레벨과 하위 레벨 처리 시 BSU에서 계산된 4x4 블록에 대한 부분 MAD들은 차례대로 이전에 저장된 값들과 덧셈기(702)에서 더해져서 쉬프트 레지스터(700)에 저장된다. 중간 레벨에서는 네 개의 부분 MAD들을 더해서 8x8 블록에 대한 완전한 MAD를 구할 수 있으며 하위 레벨에서는 열 여섯 개의 부분 MAD들을 더해서 16x16 블록에 대한 완전한 MAD를 구할 수 있다.

도 8은 본 발명에 따른 움직임 추정 장치에 대한 블록도이다. 도 8에 따른 움직임 추정 장치는 기본 탐색 유닛(800), 제1MUX(802), 비교기(804), 제2MUX(806), 어드레스 발생부(808), 인접 움직임 벡터 저장부(810), 부분 MAD 저장부(812) 및 메모리(814)를 포함한다.

기본 탐색 유닛(800)은 상술한 바와 같이 현재 프레임의 블록 데이터와 탐색영역의 데이터의 정합기준값인 MAD를 출력한다. 제1MUX(802)는 상위레벨 탐색인 경우 기본 탐색 유닛의 MAD를 출력하고, 중·하위레벨이면 도 7에 도시된 바와 같은 부분 MAD 저장부(812)의 MAD를 출력한다. 비교기(804)는 입력되는 MAD를 순차적으로 비교하여 가장 작은 MAD를 갖는 탐색점의 인덱스를 출력한다. 제2MUX(806)는 상위 또는 중간레벨이면 비교기(804)에서 출력되는 인덱스를 출력하며, 하위레벨이면 인접 움직임 벡터 저장부(810)에서 출력되는 인덱스를 출력한다. 인접 움직임 벡터 저장부(810)는 중간레벨의 초기 탐색점으로 사용되는 움직임 벡터 후보 3개중 하나를 인접 움직임 벡터들로부터 구할 때, 현재 처리중인 매크로 블록에 인접한 블록의 움직임 벡터들을 저장하고, 저장된 값들로부터 중간값을 구하며 그에 해당하는 탐색점의 인덱스를 출력한다. 어드레스 발생부(808)는 제2MUX(806)의 출력 인덱스에 따라 다음 레벨의 초기 탐색점을 계산하고, 탐색영역 데이터를 기본 탐색 유닛(800)으로 가져올 수 있도록 적절한 어드레스를 발생한다. 또한 하위레벨의 탐색인 경우 출력 인덱스로부터 움직임 벡터를 구한다. 메모리(814)는 발생된 어드레스에 해당하는 데이터들을 기본 탐색 유닛(800)에 출력한다.

### 발명의 효과

본 발명은 최소 크기의 탐색 블록과 탐색 영역을 처리하는 최소의 크기를 가지는 시스템릭 어레이로 이루어진 하나의 기본 탐색 단위를 반복적으로 이용하여 매크로 블록의 MAD를 계산하므로 처리속도가 빠르고, 칩 사이즈를 작게할 수 있다.

### (57) 청구의 범위

#### 청구항 1

원 영상을 하위레벨 영상, 원 영상에 대해 각각 가로, 세로로 1/2 크기인 영상을 중간레벨 영상 및 원 영상에 대해 각각 가로, 세로로 1/4 크기인 영상을 상위레벨 영상으로하여 현재 프레임의 각 매크로 블록의 움직임을 상기 매크로 블록에 대응하는 이전 프레임의 탐색영역과 비교하여 추정할 때, 상기 상위레벨로부터 상기 하위 레벨까지 계층적으로 탐색하는 움직임 추정 장치에 있어서,

현재 프레임 데이터와 이전 프레임 데이터를 저장하는 메모리;

상기 메모리로부터 상기 매크로 블록 데이터를 소정 크기의 블록단위로 독출하고, 상기 탐색 영역 데이터를 독출하여 독출된 데이터들의 정합 정도를 나타내는 정합기준값을 출력하는 기본 탐색 유닛;

상기 중간레벨 또는 하위레벨을 탐색하는 경우 상기 기본 탐색유닛의 출력값을 상기 매크로 블록의 전 영역에 대해 가산하여 저장하는 부분 정합기준값 저장수단;

탐색 레벨에 따라 상기 기본 탐색 유닛의 출력값과 상기 부분 정합 기준값 저장수단의 출력값중 하나를 선택하여 출력하는 제1멀티플렉서;

상기 제1멀티플렉서로부터 출력되는 값을 순차적으로 비교하여 상기 탐색영역내에서 가장 작은 정합기준값을 갖는 탐색점의 인덱스를 출력하는 비교기;

상기 하위레벨을 탐색하는 경우 현재 처리중인 매크로 블록의 인접 매크로 블록중 움직임 벡터값을 이미 알고있는 매크로 블록에 대한 움직임 벡터들의 중간값을 저장하는 인접움직임 벡터 저장부;

탐색 레벨에 따라 상기 비교기의 출력과 상기 인접움직임 벡터 저장부의 출력중 하나를 출력하는 제2멀티플렉서; 및

상기 제2멀티플렉서에서 출력되는 인덱스에 해당하는 상기 메모리의 어드레스를 발생하여 상기 메모리에

출력하는 어드레스 발생부를 포함함을 특징으로하는 움직임 추정 장치.

### 청구항 2

제1항에 있어서, 상기 기본 탐색 유닛은

상기 매크로 블록 데이터를 순차적으로 지연시키는 복수의 제1지연기;

상기 탐색영역 데이터를 소정 간격의 인덱스에 대응하여 독출하고, 소정 선택신호에 따라 각 독출된 데이터중 하나를 선택하여 출력하는 복수의 멀티플렉서;

상기 멀티플렉서들이 소정 클럭마다 순환되면서 순차적으로 동일한 독출 데이터를 출력하도록 상기 선택신호를 출력하는 복수의 제2지연기; 및

상기 지연기들의 출력과 상기 멀티플렉서들의 출력의 정합기준값을 계산하는 복수의 처리단위(PE)를 구비함을 특징으로하는 움직임 추정 장치.

### 청구항 3

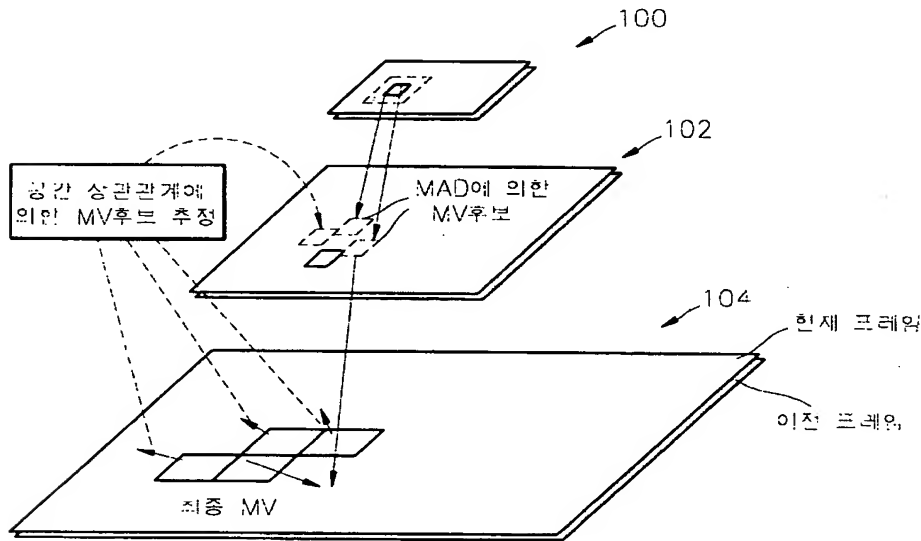
제1항에 있어서, 상기 부분 정합기준값 저장수단은

상기 기본 탐색 유닛의 출력값을 이전에 저장된 값들과 더하는 덧셈기; 및

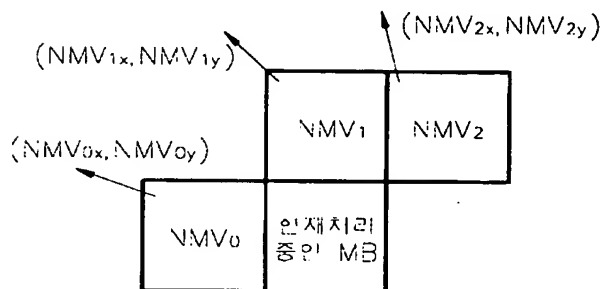
상기 탐색영역 데이터 수만큼의 크기를 갖고 상기 덧셈기의 출력을 순차적으로 저장하는 쉬프트 레지스터 어레이를 구비함을 특징으로하는 움직임 추정 장치.

### 도면

도면1

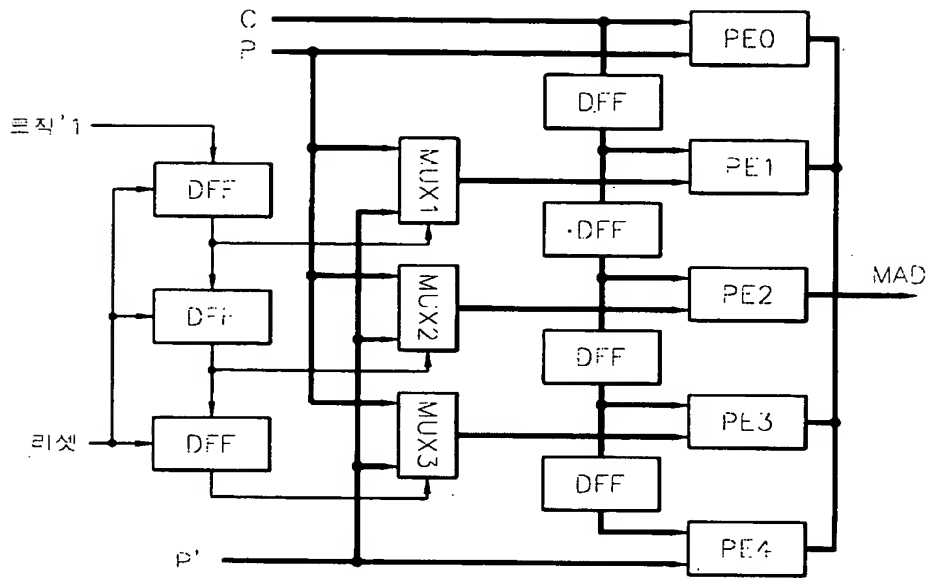


도면2

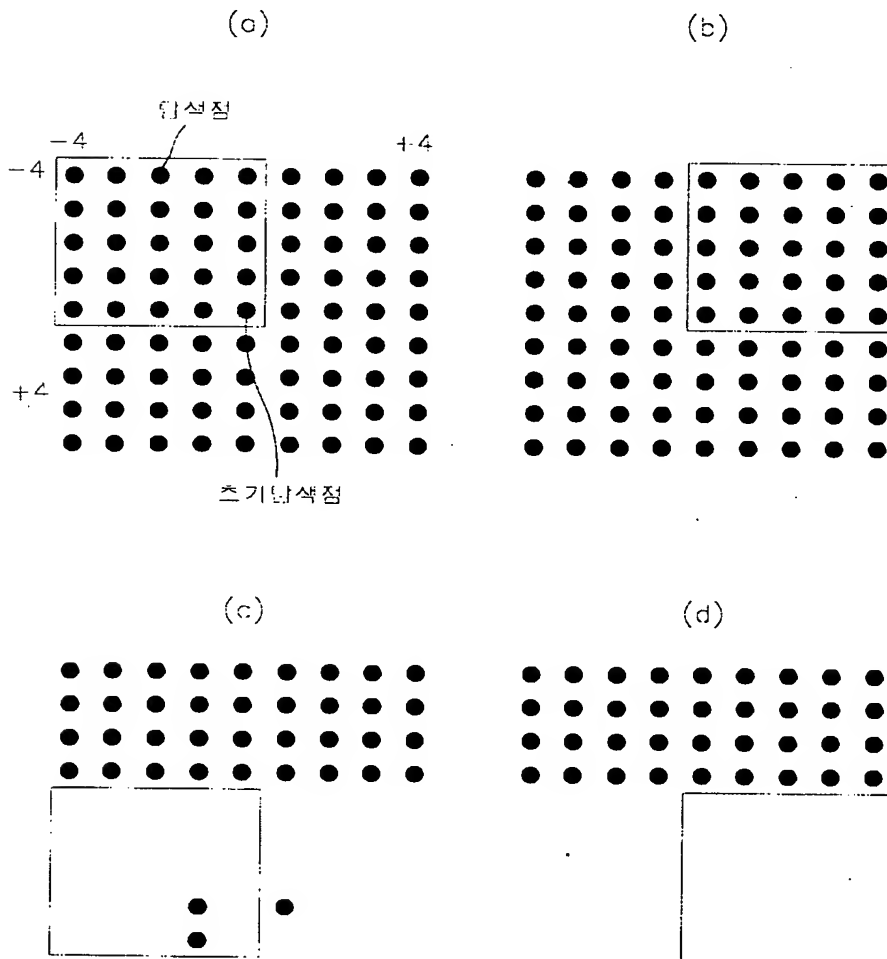




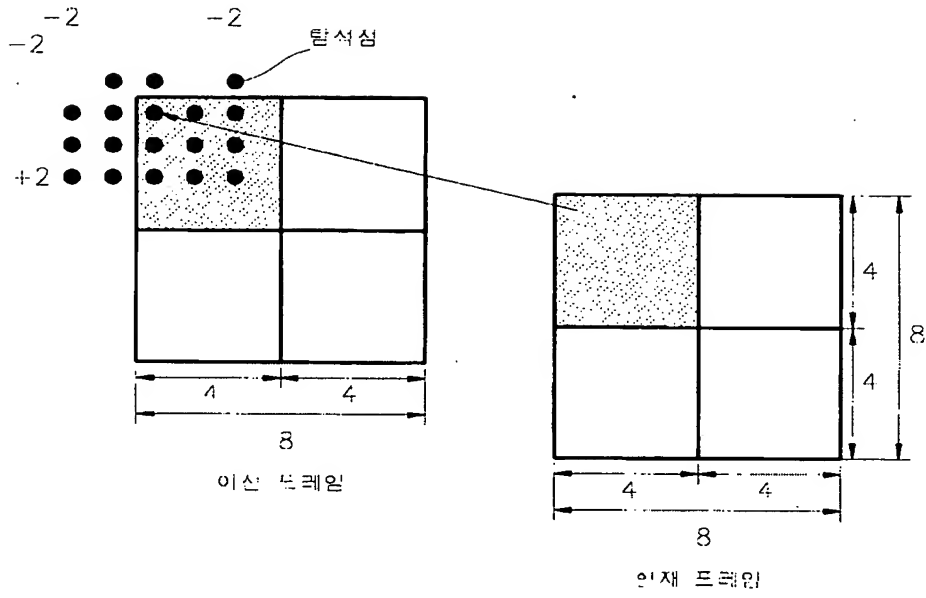
도면3



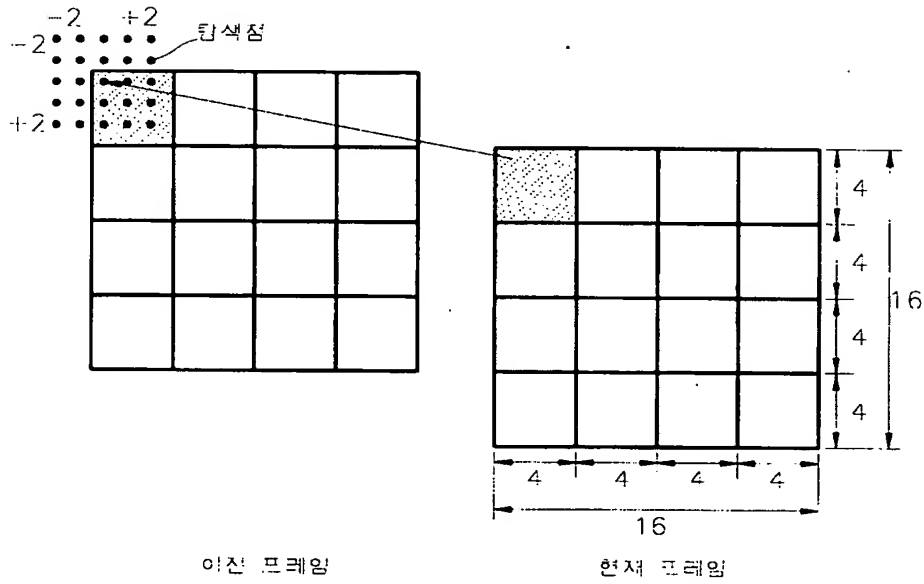
도면4



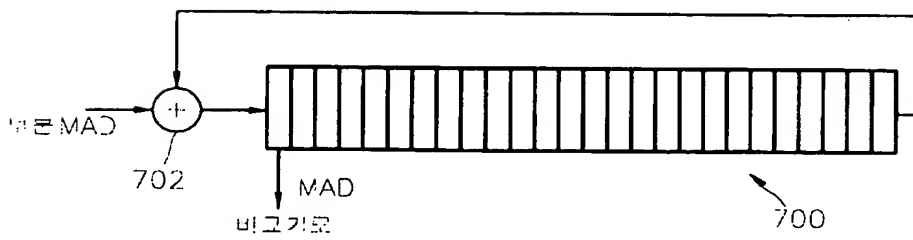
도면5

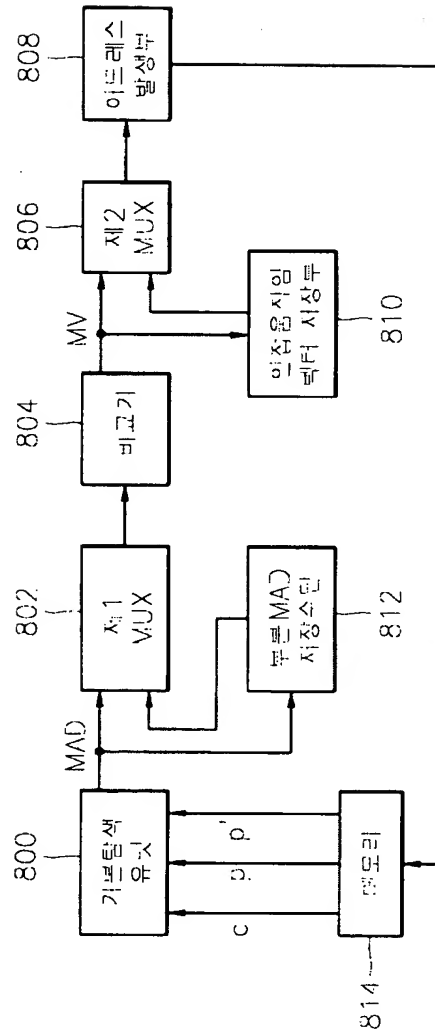


도면6



도면7





도 8